



SSE | Tutorial 1

Task 1

The programming language of the tutorial is C#. Read any C#-Tutorial of your choice¹ and create a simple Hello-World console application using Visual Studio. Visual Studio can be downloaded from the Microsoft Azure Dev Tools Portal² of TU-Chemnitz. Answer the following questions:

- What are the responsibilities of the .NET Framework?
- What is common to C#, VB.NET, F#, J#?
- What are the Lambda-Expressions and LINQ?

Task 2

- a) Get informed about Unit Testing and Test Driven Development (TDD)³. Answer the following questions:
 - What are the advantages and disadvantages of writing unit tests?
 - What is the difference between unit, integration and system tests?
 - How does the lifecycle of TDD look like?
- b) Implement a simple *Calculator* application in the TDD manner. The application should enable the following computations:
 - Multiplication of two integers
 - Division of two integers. If the second integer is 0, -9999 (meaning some error code) should be returned.

Task 3

Inform yourself about Mock Objects⁴ and when they should be used. Extend the *Calculator* class from the Task 2 to write the result of the computation to a file on a local hard drive. Use TDD and Mock Objects to simulate exceptional situations (e.g., drive is not ready or file is locked). In case of such situations -9999 should be returned by the both methods.

Task 4: Homework

Create a class for computing a hash value of a string. The hash value is the result of the division of the sum of ASCII codes of all the characters in a string by 127. Example: $\text{hash}(\text{"VSR"}) = 86 + 83 + 82 \bmod 127 = 124$. If the given string is empty, return -1. Use TDD and Unit Testing for the development.

¹ One tutorial can be found at http://www.homeandlearn.co.uk/csharp/csharp_s1p1.html

² <https://www.tu-chemnitz.de/urz/software/dreamspark.php>

³ One tutorial can be found at <http://blog.cellenza.com/publications/articles/tutorial-test-driven-development-with-visual-studio-2012/>

⁴ http://en.wikipedia.org/wiki/Mock_object